M2 Milestone Report

1. For milestone two, as planned we tried topic modelling using Latent Dirichlet Allocation (LDA) technique. For this number of topics were picked and data was tokenized. The LDA was then assigned to find 10 topics which differentiates input texts into different emotions. For instance, our system first generates a document topic probabilities. It takes a particular topic and calculates how many times the topic has occurred in our dataset. After this, the topic words distribution are analyzed by calculating conditional probability of each and every word and similar words are grouped into one topic. The library we used for topic modelling was the gensim library. Using this library, a dictionary of sentiment and corpus is created and the term frequency of a particular text is analyzed. The document is converted to a bag of words and checked whether the particular word has been repeated in the corpus and how many times it has been repeated. Based on this number, the probability was calculated. The words with equal semantics and probabilities are then grouped together under a topic. In the end, different topics are generated by comparing and matching a token pattern and categorization. While visualizing the topics, we noticed that there were still few stop words that did not add meaning for our sentiment classification. Going further we plan to remove these stopwords and use the final topics as a clue to the user's interest and preferences. We would then use it with sentiment categorization to map with recommendations.

For the sentiment analysis, using the LSTM and embedding model proved to be useful for achieving a high test accuracy. By adjusting the hyperparameters and trying out various layer dimensions, optimizers and schedulers, we were able to achieve an average of 89% accuracy for the emotions_final dataset.

2. There were no feature changes for M2

3. The current challenge stopping our sentiment analysis model from further improving is mainly overfitting. With certain optimizers and correct hyperparameters, our model's test accuracy is able to converge quickly. Yet, our test loss increases the longer we test. At the moment, this is not a high-priority problem as our model is already performing better than expected. Some approaches to counteract the overfitting are adding dropout, batch norm or removing stopwords by visualizing the input dataset. We are also likely to use early stopping for our model. Our last approximately 1000 entries for our dataset are not formatted correctly (shifted) and thus are not giving correct predictions which is lowering our test accuracy. As of now, we are not sure why this is happening, but we hope to fix it at least by the next milestone. As a last resort, we believe it is not unreasonable to cut off these last entries as we already have 22000 entries anyways which is unlikely to have a large effect on our overall result.

4. Max: Tried out different hyperparameters of our model under the "max" branch to achieve a high test accuracy. Implemented some early stopping in the training code for the model.

Sijan: Did topic modelling to the input dataset using gensim library. Analyzed topic words distribution, term frequency and created a dictionary of sentiment and corpus. After this converted the inputs to a bag of words and grouped words with equal semantics and probabilities under the same topic.

Kaleb: Trained the sentiment analysis model trying various hyperparameters. Adjusted the different dimensions for the layers in the model as well as trying out different optimizers and schedules. Also verified the results from the model by printing out a certain number of correct and incorrect outputs and checking if they seemed accurate.

Ryan: Basic implementation of the GUI